
UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte JOCHEN KAPPEL,
JOSEF MARKGRAF,
and MICHAEL MEADOWS

Appeal 2007-0226
Application 09/823,866[1]
Technology Center 2100

Decided: June 27, 2007

Before JAMES D. THOMAS, LEE E. BARRETT, and JAY P. LUCAS, *Administrative Patent Judges*.

BARRETT, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the final rejection of claims 1-22. We have jurisdiction pursuant to 35 U.S.C. § 6(b).

We affirm-in-part.

---

[1] Application for patent filed March 31, 2001, entitled "Object to Object Communication System and Method," now U.S. Pub. 2002/0052979 A1, published May 2, 2002.

BACKGROUND

The claims are directed to providing object to object communication in a networking environment.

Claim 1 is illustrative:

> 1. A system for providing object to object communication, comprising:
>
> means for identifying at least two objects in separate and distinct server locations from a plurality of objects to communicate;
>
> means for locating the at least two objects to communicate; and
>
> means for using a component framework to enable the communication of the at least two objects.

THE REFERENCES

Konrad	US 5,544,320	Aug. 6, 1996
Foody	US 5,732,270	Mar. 24, 1998

Douglas C. Schmidt, *Wrapper Facade: A Structural Pattern for Encapsulating Functions within Classes*, C++ Report Magazine, February 1999, pages 1-10.

THE REJECTIONS

Claims 1-4, 6-9, 11-14, 16-19, 21, and 22 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Schmidt and Konrad.

Claims 5, 10, 15, and 20 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Schmidt and Konrad, further in view of Foody.

Claims 11-15 stand rejected under 35 U.S.C. § 101 as being directed to nonstatutory subject matter. This is a new ground of rejection in the Examiner's Answer.

DISCUSSION

35 U.S.C. § 101: Claims 11-15

The Examiner entered a new ground of rejection of claims 11-15 under 35 U.S.C. § 101 in the Examiner's Answer. A new ground of rejection is permitted in an examiner's answer. *See* 37 C.F.R. § 41.39(a)(2); *Manual of Patent Examining Procedure* (MPEP) § 1207.03 (8th ed., Rev. 5, Aug. 2006). The Examiner properly gave notice of the new ground of rejection (Answer 4, 11-12) and obtained approval of the Technology Center Director (Answer 12). *See* MPEP § 1207.03. In response to a new ground of rejection, an appellant must either file a request to reopen prosecution, 37 C.F.R. § 41.39(a)(2)(b)(1), or a reply, § 41.39(a)(2)(b)(2), to avoid a *sua sponte* dismissal of the appeal as to the claims subject to the new ground of rejection. Appellants have not taken either action. Accordingly, the rejection of claims 11-15 is affirmed.

35 U.S.C. § 103(a)

Claims 1-4, 6-9, 11-14, 16-19, 21, and 22

Appellants do not argue the separate patentability of the claims. Therefore, the claims in this group stand or fall together with representative claim 1. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Claim 1 is in means-plus-function format. Appellants' brief does not correspond the means to the structure in the specification as required by 37 C.F.R. § 41.37(c)(1)(v) and, therefore, we assume the means can be any structure for performing the function.

Rejection

Initially, we try to clarify the rejection. The Examiner finds (Final Rejection 2): "Regarding claims 1-22, it is noted that as disclosed, an object refers to a function/procedure and a component to a set of objects. See application as filed, page 10, lines 9-10." The Specification states:

> In an object-based model of components, a component can be seen as a stateless object that provides a set of objects. These objects are for the most part, similar to everyday functions or procedures.

Page 10, ll. 8-10. This is the closest the Specification comes to defining an "object." Relevant to an understanding of the rejection is the fact that Appellants' disclosed invention uses "wrapper facades" and Schmidt is directed to "wrapper facades." The Specification describes:

> If it is determined at step 34 that the objects are in different components, the object to object communication system then uses a wrapper facade to facilitate the object to component communication at step 35. Wrapper facades are an encoding of information to allow for object to component communication.

Page 15, ll. 2-6. That is, wrapper facades allow access to objects.

The Examiner finds that "[a]s to claim 1, Schmidt teaches a system for providing object to object communication (client - server communications)

(Final Rejection 2). This statement seems to say that the Examiner considers a "client" and "server" to be "objects," instead of functions and procedures on the client and server. However, since Schmidt describes "wrapper facades" whose purpose is "to encapsulate low-level functions and data structures with object-oriented (OO) class interfaces" (Schmidt, p. 1 under "Introduction") for use in networking applications (Schmidt, p. 3 under "Problem" and "Solution"), i.e., encapsulation of objects, we interpret the rejection, in context, to be that the "wrapper facade" in Schmidt provides access to objects on the clients and server in the networking environment.

The Examiner continues by finding that Schmidt teaches "means for identifying at least two objects (one being the client and one being database/printer service) from a plurality of objects (client, database, printer, console services/functions) to communicate (invoke/request service) [fig.s [sic] 1, 3" (Final Rejection 2). Again, while this seems to say that the physical objects of client computer, database, printer, and console in Figure 1 are "objects," in the context of Schmidt, we interpret the rejection to be that "objects," in the sense of functions or procedures, on the client computer, database, printer, etc. in Figure 1 are in communication.

The Examiner further finds that Schmidt "means for locating the at least two objects to communicate (socket handles) [page 1, right col.; page 6, right col., 2nd code listing" (Final Rejection 2). The two objects (one on the client and one on the server) communicate via "socket handles" which are ports at a certain IP address.

The Examiner lastly finds that Schmidt teaches "means for using a component framework (wrapper façade implemented as frameworks such as ACE) to enable the communication (forward client invocations) of the at least two objects [page 4, sections 2.7, 2.8; page 6, section 'The socket wrapper façade']" (Final Rejection 2).

As to the limitation of "two objects in separate and distinct server locations" in claim 1, the Examiner finds (Final Rejection 2-3): "[I]n a client/server configuration, a client request[s] a service and the server provides the service. An object is a client to one object and is a server to another object." The Examiner finds that Konrad teaches a client/server relationship between objects (*id.* at 3). The Examiner concludes that "[w]hen the teachings are combined, a client machine of Schmidt would have behaved as both a client machine/host and a server machine/host, and therefore the two communicating objects would have been located on separate and distinct server locations/machines" (*id.*).

While the Examiner's application of Konrad is not exactly clear, especially the client/server discussion, Konrad discloses communication between a local host (which can be a multi-user system, i.e., a server) and a remote host (which is a server). Both hosts have objects that are in communication. We interpret the rejection to be that it would have been obvious for the client in Schmidt to be a server or to add a server in Schmidt in view of the server to server communication in Konrad.

Content of Schmidt and Konrad

Schmidt describes "wrapper facades" whose purpose is "to encapsulate low-level functions and data structures with object-oriented (OO) class interfaces" (Schmidt, p. 1 under "Introduction") for use in networking applications (Schmidt, p. 3 under "Problem" and "Solution"). "The *Functions* are existing low-level functions and data structures that provide a cohesive service." Page 4 § 2.6. "The *Wrapper Facade* is a set of one or more classes that encapsulate the Functions and their associated data structures. The Wrapper Facade provides methods that forward client invocations to one or more of the low-level Functions." *Id.* Either the functions or the wrapper facade can be considered to be "objects." The implementation applies reusable components from the ACE framework. "ACE provides a rich set of reusable C++ wrappers and framework components that perform common communication software tasks across a wide range of OS platforms." Page 4 § 2.8. Thus, the ACE framework corresponds to "means for using a component framework to enable the communication of the at least two objects." Schmidt provides an example of a logging server handling connection requests and logging record requests sent by clients to a client connection endpoint called a "socket handle," which is a port at a certain IP address. Page 1 § 2.2. Since the wrapper facade is used at the client and the logging server, and because there is communication across a wide range of OS platforms using the ACE framework, there is "object to object communication" across a network.

Schmidt has to have "means for identifying at least two objects . . . from a plurality of objects to communicate" because it must know to connect the logging record sent by the client to the program on the server which processes the records. Schmidt must also have "means for locating the at least two objects to communicate" in order to be able to connect the object on the client with the object on the logging server.

Schmidt does not expressly teach "two objects in separate and distinct server locations." Note that this limitation does not recite "objects that reside on separate servers" (Br. 11), as argued by Appellants; technically, the "separate and distinct server locations" could be in different address locations in the same server. A "server" is usually defined as a computer system in a network that is shared by multiple users. Stand-alone PCs can function as a server to other users on the network even though they serve as a single workstation to one user. The logging server in Schmidt is clearly a server. Schmidt shows communication from a "client" to the server. A "client" is a computer that requests a service from a server. Although the client computers in Schmidt could serve multiple users, and thus be a server, this is not disclosed.

Konrad discloses a system which allows a computer user at a local host to access information services on a remote host which are as easy to access as services on the local host (e.g., col. 4, ll. 6-14). In a "client-server-service (CSS)" model, a "client" process makes demands on a "server" process, which then satisfies these demands using a "service" process (col. 6,

ll. 19-23). If the client on the local host desires some data on a remote host, the client sends a request for service to a server process on a remote host, the server accepts the request and conveys it to a database service, the server retrieves the response from the database service on the remote host and returns the response to the requesting client (col. 6, ll. 24-39). The "local host" may be a "local multi-user system" (col. 4, ll. 9-10). A client is a process which issues requests (col. 7, ll.33-34). A "server" is an intermediary between a client and a service (col. 7, ll. 35-36), such as between the Human Interface Server between a Remote Object Client and a Human Interface Service and a Starter Server between a Starter Client and a Starter Service, as shown in Figure 2. The local host and remote hosts can be at different locations (col. 8, 21-56).

Analysis

Claim 1 is extremely broad and essentially recites a system for enabling two objects at different server locations (not necessarily different servers) to communicate. The term "object" is a broad term that is not expressly defined in the Specification, except loosely that the "objects are for the most part, similar to everyday functions or procedures" (Specification 10, ll. 9-10). Thus, an object could be any program and is not limited to a "class" in an objected-oriented programming language. "Server" is also not defined. We presume Appellants intend the conventional meaning of server as a computer system in a network that is shared by multiple users. Since server-to-server communication is notoriously well known in the Internet

age, and since there must be a program (object) on each server to communicate, it is not apparent why the claims do not cover known server-to-server communications. If Appellants intend some special meaning for the claim terms, it has not been set forth in the claims or expressly defined in the Specification. Nevertheless, we have to address the rejection before us.

Since Schmidt discloses wrapper facades it discloses "objects." Since Schmidt discloses communication over a network it necessarily teaches one skilled in the art that there are "objects" at both ends of the communication. Schmidt does not expressly teach "two objects in separate and distinct server locations" because it shows an example of a plurality of "clients" in communication with a "logging server" over a network. Nevertheless, Schmidt is not limited to this illustrative example. We understand the Examiner's rejection to be that the clients in Schmidt could be servers, or that a server could be added, in view of Konrad. Since Konrad, Figure 2, shows two hosts in communication, where the local host can be a local multi-user system, i.e., a server, and the remote host is a server for several local hosts, Konrad discloses server to server communication. Konrad's definition of server is different than Appellants' usage of the term. It may be that Konrad alone could meet the terms of claim 1 since the programs on the hosts can be considered objects. However, one skilled in the art would have known to apply the teaching of server to server communication in Konrad to Schmidt. The only difference is that the client in Schmidt serves a single user, instead of as a server that is shared by several users. The motivation is

that it is often necessary to communicate between servers in a networking environment. Therefore, we conclude that the combination of Schmidt and Konrad establishes a prima facie case of obviousness.

Appellants argue that the claims recite "objects that reside on separate servers" and that "Applicants are not claiming that their system, method or computer readable medium provides objects that 'act like' both servers and clients, but rather that there are objects residing on servers" (bolding omitted) (Br. 11). The Examiner repeats the rejection (Answer 10).

The rejection could have been better stated. However, we interpret the rejection, in the context of the teachings of Schmidt and Konrad, to be that it would have been obvious that the network example of Figure 1 of Schmidt could include another server, or that one of the clients could be a server, in view of the server to server communication taught by Konrad. We do not think Appellants can reasonably dispute that server to server communications were notoriously well known in the networking art. Since the wrapper facades in Schmidt correspond to the claimed objects, and are used in elements at both ends of a communication, the objects are on the networking elements. Therefore, if a client in Schmidt was a server, it would use wrapper facades (objects) for communication with other servers.

Appellants argue (Br. 11-12):

> [Obviousness requires] that references and motivation be provided to show, at a minimum, that it would be obvious to add at least one more server to Schmidt, not that it would be "obvious" to re-define the role of an object as "functioning" as a server and client simultaneously. The practice of redefining an object so that it "functions" as both a

> client and a server is simply not the same as providing objects residing on servers.

As discussed, we interpret the rejection to be that it would have been obvious to add a server in Schmidt. Example 1 of Schmidt is simply an illustrative embodiment of where facade wrappers would be used in a networking environment. One of ordinary skill in the computer communications art would have been taught by Konrad, if not from her own basic knowledge in the field, that servers communicate among each other.

For the reasons stated above, we conclude that the references establish a prima facie case of obviousness which has not been rebutted. The rejection of claims 1-4, 6-9, 11-14, 16-19, 21, and 22 is affirmed.

Claims 5, 10, 15, and 20

The Examiner admits that Schmidt does not disclose translation from one view to another view. The Examiner referred to the rejection of claim 4 for "address classes," where it is stated that Schmidt teaches using objects to represent functions such as threading, sockets, and mutex and concluded that "it would have been obvious to also represent address related functions by corresponding objects/classes" (Final Rejection 4). The Examiner found that "Foody teaches object communication across heterogeneous systems (fig. 11), including translating from one view to another view (convert types) during communication (call)" (*id.*) and concluded that it would have been obvious to include such translation from one view to another view in Schmidt to provide bi-directional interoperability (*id.*).

Appellants argue that the Examiner's "statement regarding combination and alleged motivation does not address the issue of limiting translation from one view to another if the at least two objects are address classes, as specifically recited in each of the subject pending claims" (Br. 13). Appellants also argue that since Schmidt already provides bi-directional interoperability and there is no reason to add it (*id.*).

The Examiner's Answer does not respond to these arguments.

We will not sustain the rejection. The Examiner has not shown translation of "address classes" from one view to another. The rejection concludes that "it would have been obvious to also represent address related functions by corresponding objects/classes" (Final Rejection 4) and then concludes that it would have been obvious to provide translation. It is not clear that Foody discloses translation from one view to another, but, if so, there appears to be no reason why one skilled in the art would have had a reason to make such modifications to provide bi-directional interoperability. Accordingly, the rejection of claims 5, 10, 15, and 20 is reversed.

It is not clear that the limitation that "the at least two objects are address classes" in the instant claims is accurate. The Specification describes that if there are no exported classes, "[t]he use of objects that address classes must then employ some form of translation from one view to another" (page 12, ll. 23-24). This seems to say that "objects address classes" ("address" used as a verb) rather than "objects are address classes" ("address" used as an adjective). Rather than enter a new ground of

rejection, we leave it to Appellants and the Examiner to sort out whether the language is accurate.

CONCLUSION

The rejection of claims 11-15 under 35 U.S.C. § 101 is affirmed.

The rejection of claims 1-4, 6-9, 11-14, 16-19, 21, and 22 under 35 U.S.C. § 103(a) is affirmed.

The rejection of claims 5, 10, 15, and 20 under 35 U.S.C. § 10(a) reversed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED-IN-PART

KIS

DORITY & MANNING, P.A.
POST OFFICE BOX 1449
GREENVILLE, SC 29602-1449